Motor Control Blockset™

Sensorless Field-Oriented Control (FOC) of PMSM Using Renesas® MCK-RA6T2 Motor Control Kit

MATLAB®

R2024a

MathWorks®

How to Contact MathWorks

Latest news:	`www.mathworks.com`
Sales and services:	`www.mathworks.com/sales_and_services`
User community:	`www.mathworks.com/matlabcentral`
Technical support:	`www.mathworks.com/support/contact_us`
Phone:	508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Sensorless Field-Oriented Control (FOC) of PMSM Using Renesas® MCK-RA6T2 Motor Control Kit

Revision History

June 2024	First printing	"Release for R2024a"

Contents

About This Example

Introduction

This example implements V/F (open-loop or scalar) and field-oriented control (FOC) techniques to control the speed of a three-phase permanent magnet synchronous motor (PMSM). The FOC speed control algorithm that the example implements uses the Extended EMF Observer block to obtain realtime rotor position using sensorless position estimation.

In addition to simulation of the V/F and FOC control algorithms, the example supports deployment of these algorithms on the Renesas MCK-RA6T2 motor control kit.

You can use the kit to perform motor control evaluation and debugging safely with a communication board that isolates PC and CPU board electrically. The kit has these features:

- RA6T2 motor control MCU
- Inverter board for 3-phase BLDC motor (with 48V/10A rating)
- On-board debugger for flash programming of MCU
- Overcurrent detection
- Support for one-shunt/three-shunt current sensing
- Support for position sensors

For more information about this kit, see MCK-RA6T2 Renesas Flexible Motor Control Kit for RA6T2 MCU Group.

Hardware Requirements

This section lists the hardware requirements to run the example:

1 Renesas MCK-RA6T2 motor control kit (RTK0EMA270S00020BJ) that includes these peripherals:

- Inverter board (RTK0EM0000B12020BJ)
- Controller board (RTK0EMA270C00000BJ)
- Brushless DC (BLDC) motor (R42BLD30L3) with rated voltage of 36 V and rated current of 1.67 A
- USB cable

2 24V power supply

For detailed specifications of the MCK-RA6T2 motor control kit, see MCK-RA6T2 User's Manual.

Software Requirements

This section lists the software products from MathWorks® and Renesas that you need to simulate and run the example models on the MCK-RA6T2 motor control kit.

Required MathWorks Products

To simulate the example model:

- MATLAB®
- Simulink®
- Motor Control Blockset
- Stateflow®

To generate code and deploy the example model:

- MATLAB
- Simulink
- Motor Control Blockset
- Stateflow
- Embedded Coder®
- Simulink Coder™
- MATLAB Coder

Required Renesas Products

- e^2 studio (an integrated development environment (IDE) for Renesas MCUs)

Contents of Downloaded ZIP Folder

The ZIP folder that you downloaded from the GitHub® repository, includes:

1 MATLAB project package for V/F and FOC control algorithms, which includes the following files and folders:

Folder Name	Description
`code`	Contains generated code (if any).
`components`	Contains Simulink model components for V/F and FOC control algorithms.
`data`	Contains Simulink data dictionary file (`motorControlData.sldd`).
`model`	Contains Simulink models for V/F and FOC algorithms that support both simulation and code generation.
`resources`	Contains MATLAB project resource data and logs.
`script`	Contains initialization and utility scripts to use the Simulink models.
`shortcuts`	Contains shortcut utility scripts for the MATLAB project window.
`work`	Contains any additional data or utility files (if any).

File Name	Description
`McbRenesas.prj`	File that launches the MATLAB project window.

2 Driver package for the MCK-RA6T2 motor control kit, which includes the following folder:

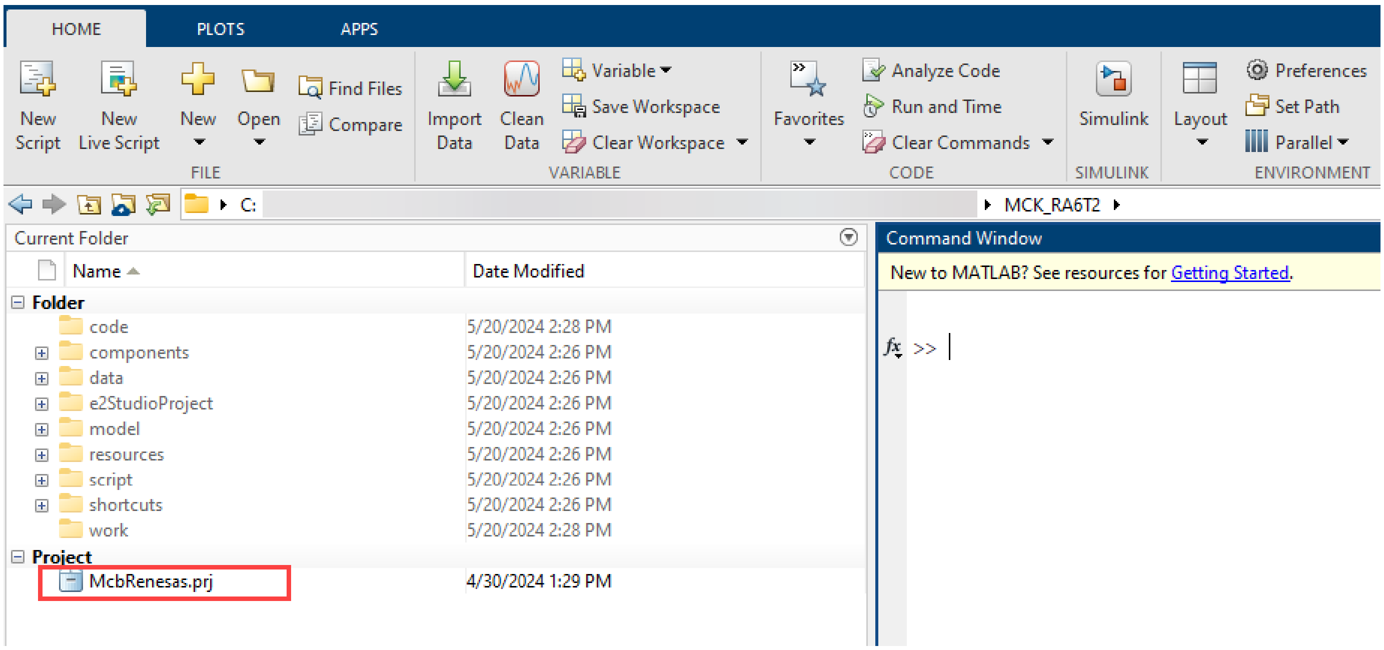
Folder Name	Description
`e2StudioProject`	Contains the control algorithm code along with driver package for the MCK-RA6T2 motor control kit (RTK0EMA270S00020BJ). The software e^2 studio uses this folder to launch the Renesas hardware project for MCK-RA6T2 motor control kit. The `src` directory available inside this folder contains the control algorithm code generated by the MATLAB project window.

V/F (Open-Loop or Scalar) Control of PMSM Using Renesas Hardware

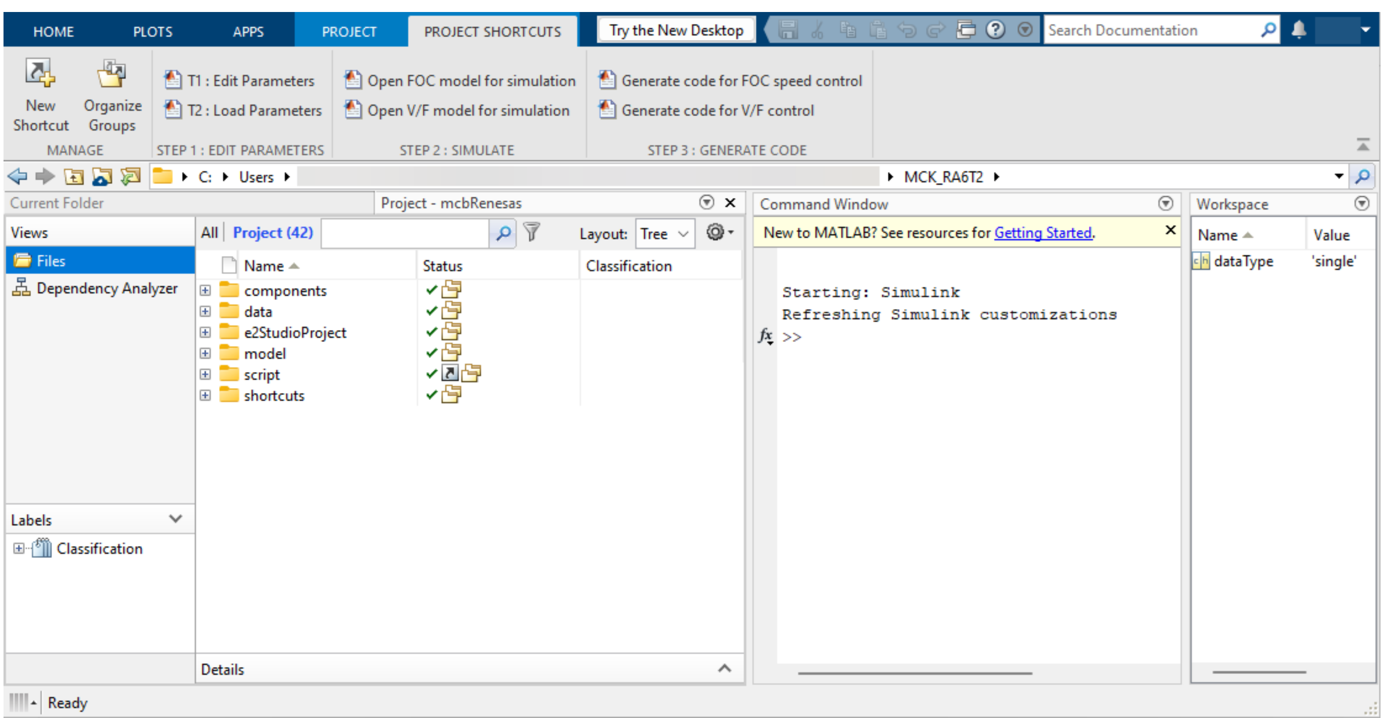
Open MATLAB Project

Follow these steps to open the MATLAB project associated with this example:

1. Unzip the `MCK_RA6T2.zip` folder that you downloaded from the GitHub repository.
2. Open MATLAB and navigate to the folder you unzipped in step 1.



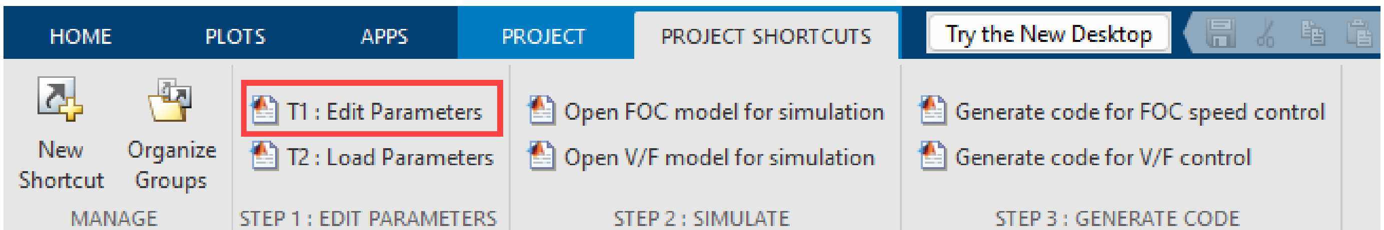
3. Double-click the file `McbRenesas.prj` to open the following MATLAB project window.



Update Motor and Inverter Parameters

The prerequisite to simulating or deploying the Simulink model containing V/F control algorithm is to edit the motor and inverter parameters and load them to MATLAB base workspace. Follow these steps to update and load the parameters:

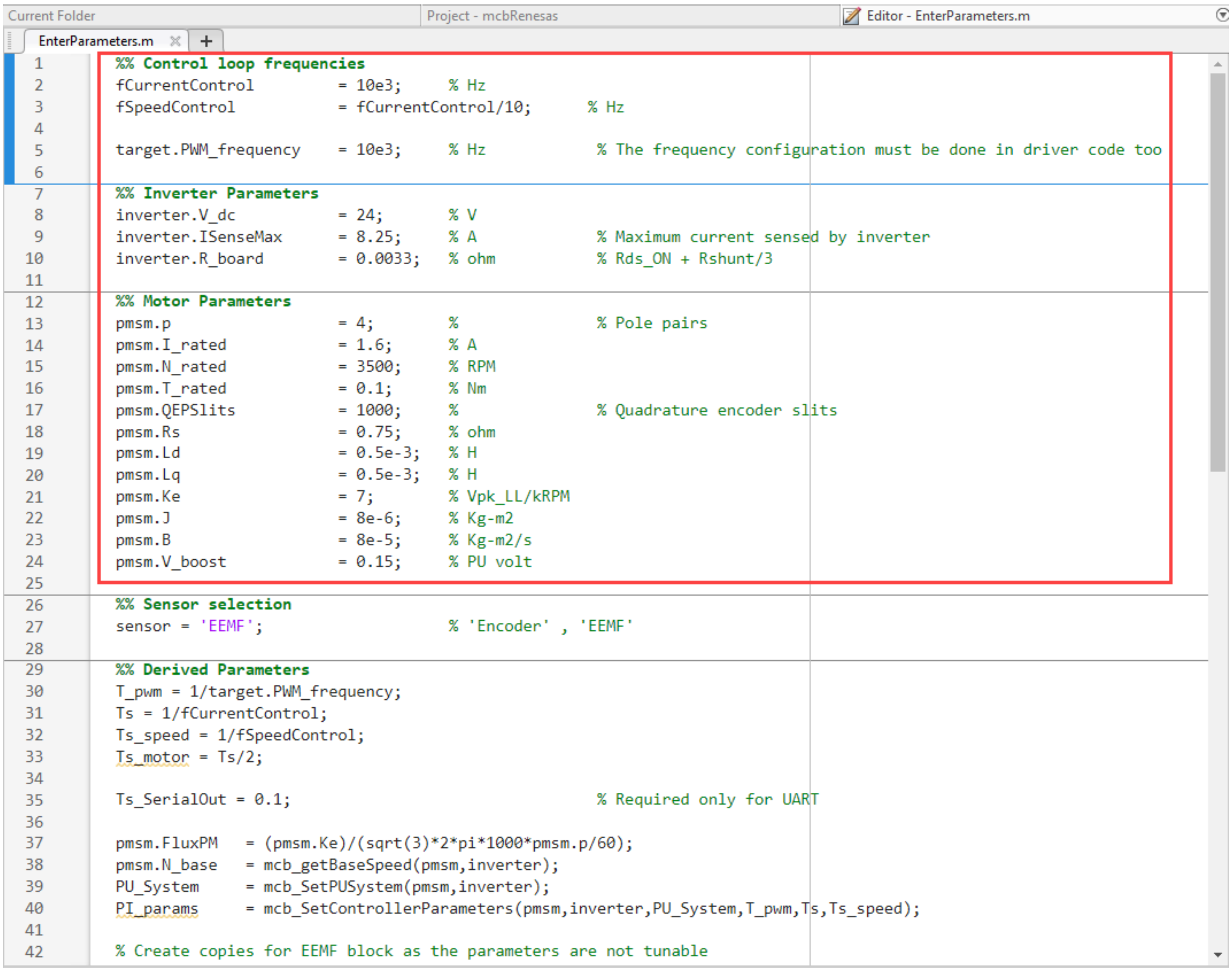
1. In the **Project Shortcuts** tab of the MATLAB project window, click the **T1: Edit Parameters** shortcut button to open the model initialization script `EnterParameters.m`.



   The Simulink model uses this script to obtain the motor, inverter, and other parameter values, which it uses to compute the derived parameters.

2. Update the following sections of the model initialization script `EnterParameters.m`:

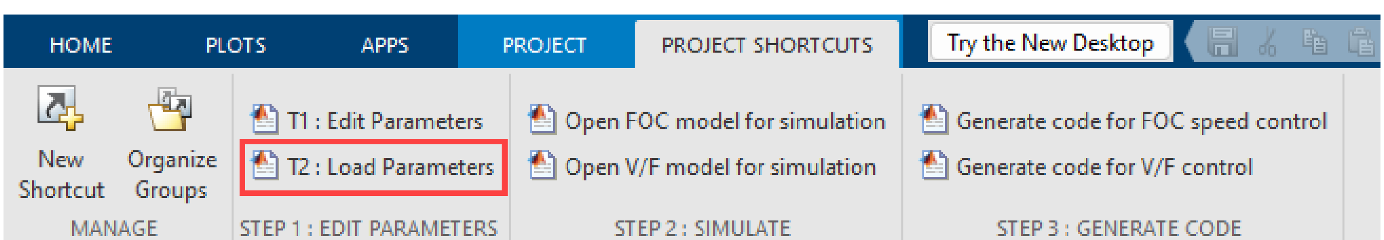
   - **Control Loop Frequencies**
   - **Inverter Parameters**
   - **Motor Parameters**

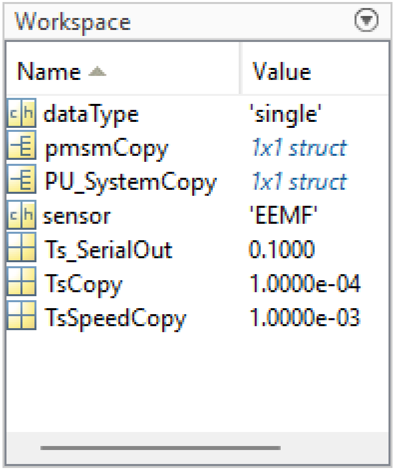



```
%% Control loop frequencies
fCurrentControl         = 10e3;     % Hz
fSpeedControl           = fCurrentControl/10;      % Hz

target.PWM_frequency    = 10e3;     % Hz           % The frequency configuration must be done in driver code too

%% Inverter Parameters
inverter.V_dc           = 24;       % V
inverter.ISenseMax      = 8.25;     % A            % Maximum current sensed by inverter
inverter.R_board        = 0.0033;   % ohm          % Rds_ON + Rshunt/3

%% Motor Parameters
pmsm.p                  = 4;        %              % Pole pairs
pmsm.I_rated            = 1.6;      % A
pmsm.N_rated            = 3500;     % RPM
pmsm.T_rated            = 0.1;      % Nm
pmsm.QEPSlits           = 1000;     %              % Quadrature encoder slits
pmsm.Rs                 = 0.75;     % ohm
pmsm.Ld                 = 0.5e-3;   % H
pmsm.Lq                 = 0.5e-3;   % H
pmsm.Ke                 = 7;        % Vpk_LL/kRPM
pmsm.J                  = 8e-6;     % Kg-m2
pmsm.B                  = 8e-5;     % Kg-m2/s
pmsm.V_boost            = 0.15;     % PU volt

%% Sensor selection
sensor = 'EEMF';                    % 'Encoder' , 'EEMF'

%% Derived Parameters
T_pwm = 1/target.PWM_frequency;
Ts = 1/fCurrentControl;
Ts_speed = 1/fSpeedControl;
Ts_motor = Ts/2;

Ts_SerialOut = 0.1;                                % Required only for UART

pmsm.FluxPM   = (pmsm.Ke)/(sqrt(3)*2*pi*1000*pmsm.p/60);
pmsm.N_base   = mcb_getBaseSpeed(pmsm,inverter);
PU_System     = mcb_SetPUSystem(pmsm,inverter);
PI_params     = mcb_SetControllerParameters(pmsm,inverter,PU_System,T_pwm,Ts,Ts_speed);

% Create copies for EEMF block as the parameters are not tunable
```

3. Click **Save** to save the file `EnterParameters.m`.
4. Switch to the MATLAB project window and navigate to the **Project Shortcuts** tab.
5. Click the **T2: Load Parameters** shortcut button to:

   - Compute derived parameters using the data available in the model initialization script.
   - Update all parameters to the Simulink data dictionary `motorControlData.sldd`.
   - Load necessary parameters from Simulink data dictionary to the MATLAB base workspace.



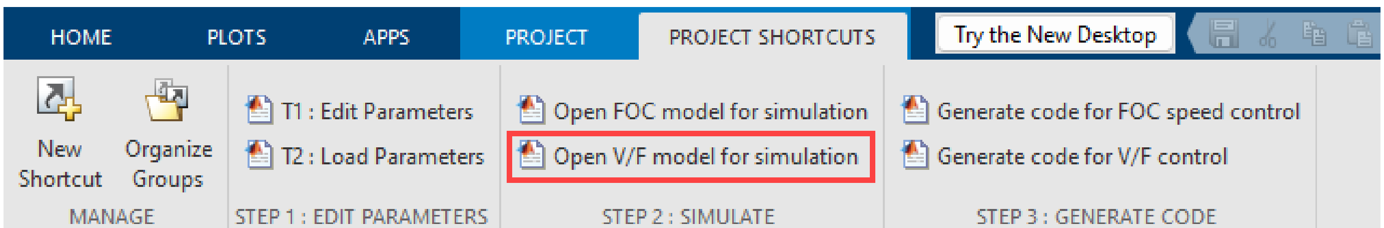


Note The shortcut **T2: Load Parameters** loads only the necessary parameters to the base workspace. The Simulink model reads the other parameters directly from the Simulink data dictionary.

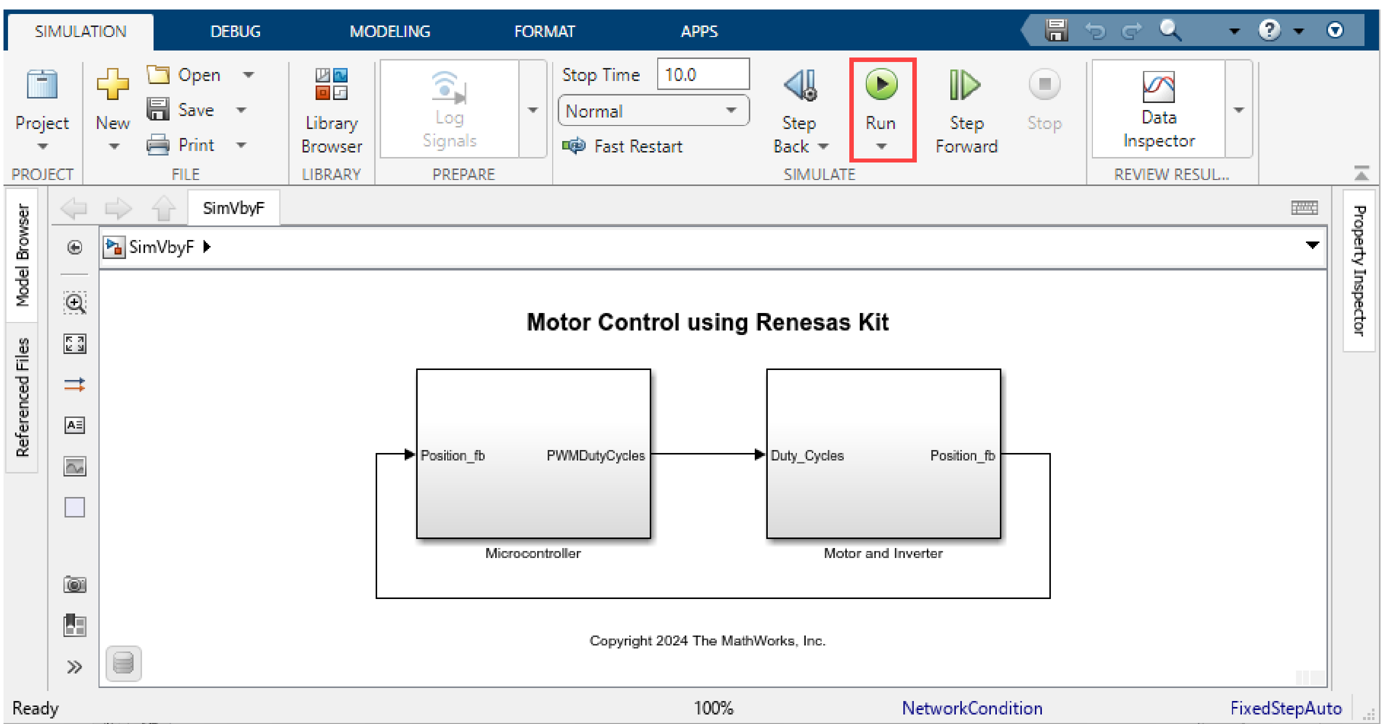
Simulate Model Containing V/F Control Algorithm

This example supports simulation. Follow these steps to simulate the Simulink model containing the V/F control algorithm.

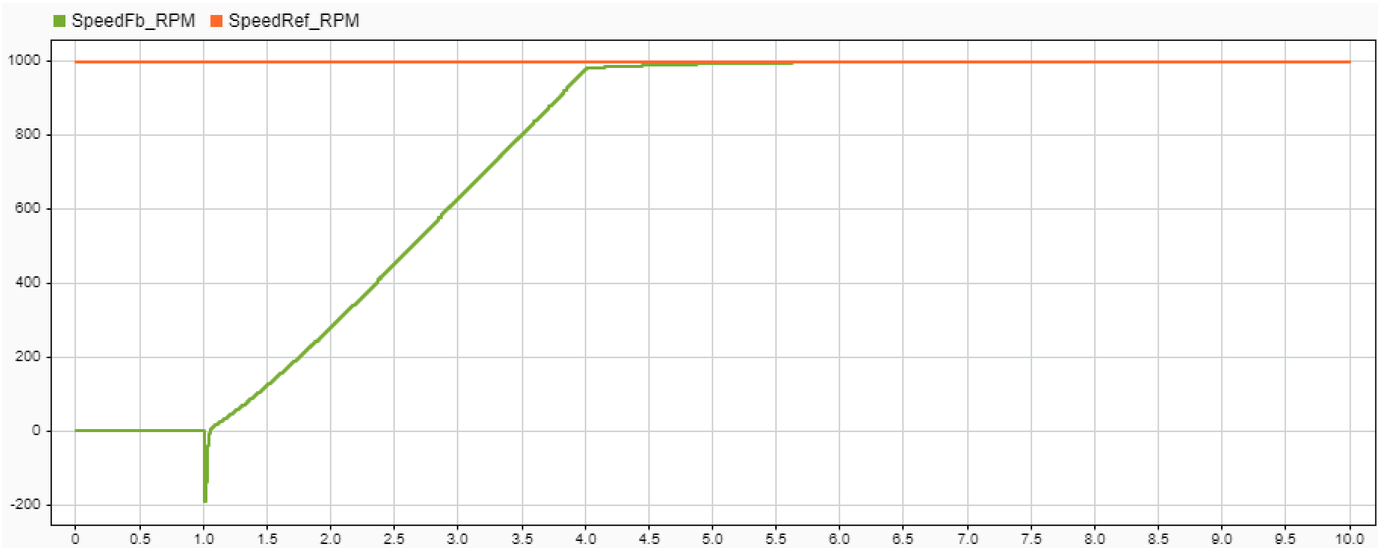
1. Ensure that you edit the motor and inverter parameters and load them to the MATLAB base workspace. For details, see the section Update Motor and Inverter Parameters.
2. In the **Project Shortcuts** tab of the MATLAB project window, click the **Open V/F model for simulation** shortcut button to open the Simulink model `SimVbyF.slx` containing the V/F control algorithm.



3. Click **Run** on the **Simulation** tab to simulate the model.



4. Click **Data Inspector** on the **Simulation** tab to view and analyze the simulation results.

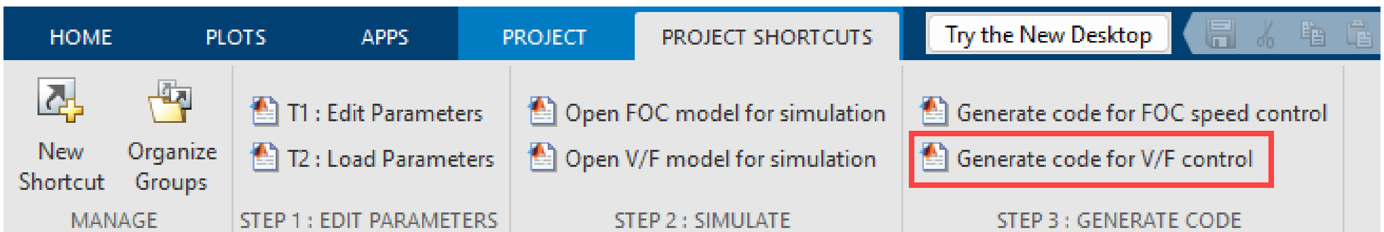

SpeedFb_RPM
SpeedRef_RPM
1000
800
600
400
200
0
-200
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0

Generate Code for V/F Control Algorithm

This section explains how to use the Simulink model `SimVbyF.slx` to generate code for the V/F control algorithm.

Follow these steps to generate the code:

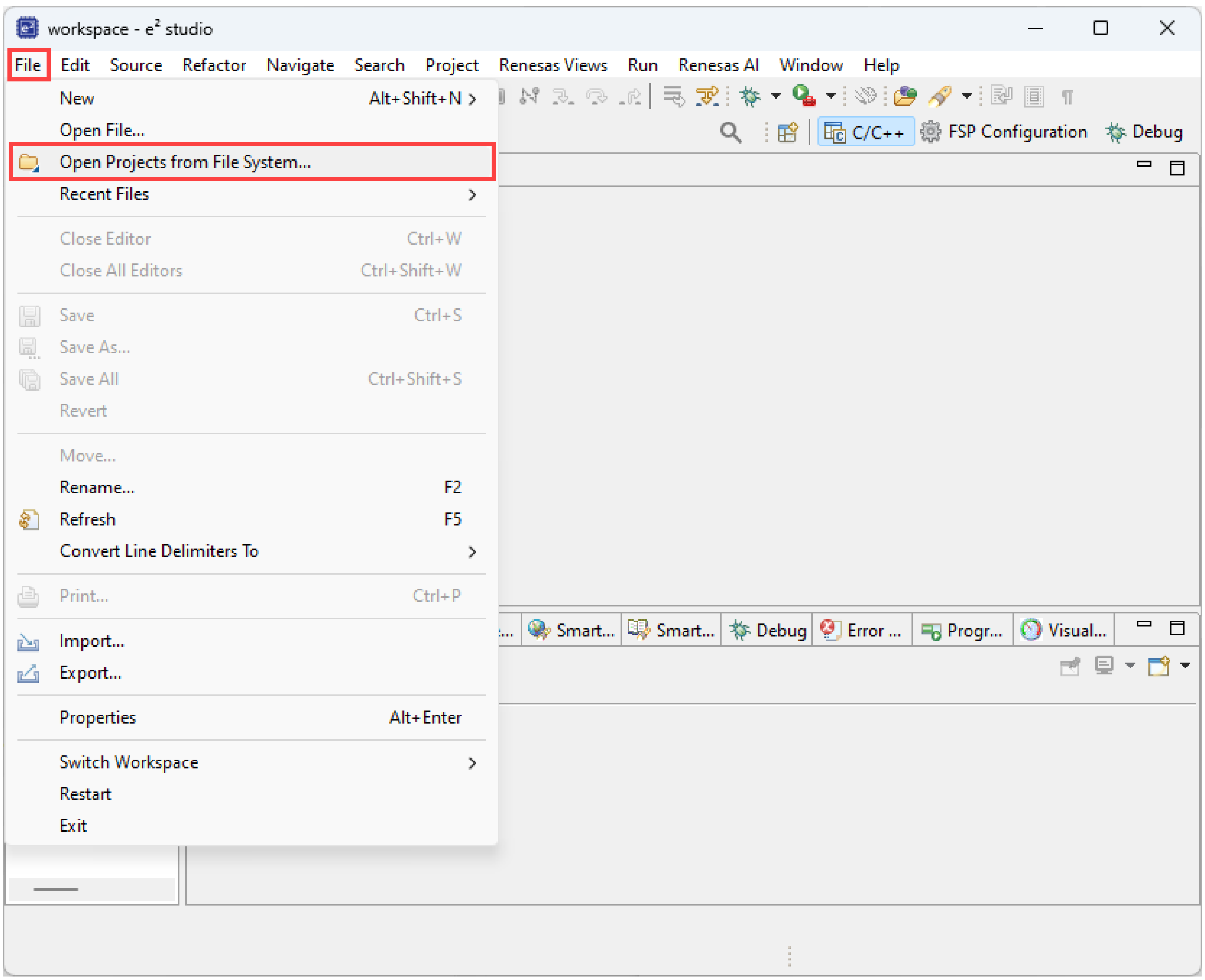
1. Simulate the model `SimVbyF.slx` and observe the simulation results. For details, see the section Simulate Model Containing V/F Control Algorithm.
2. Click the **Generate code for V/F control** shortcut button to build the model `SimVbyF.slx` as well as generate and save code inside `src` directory available in the `e2StudioProject` folder.



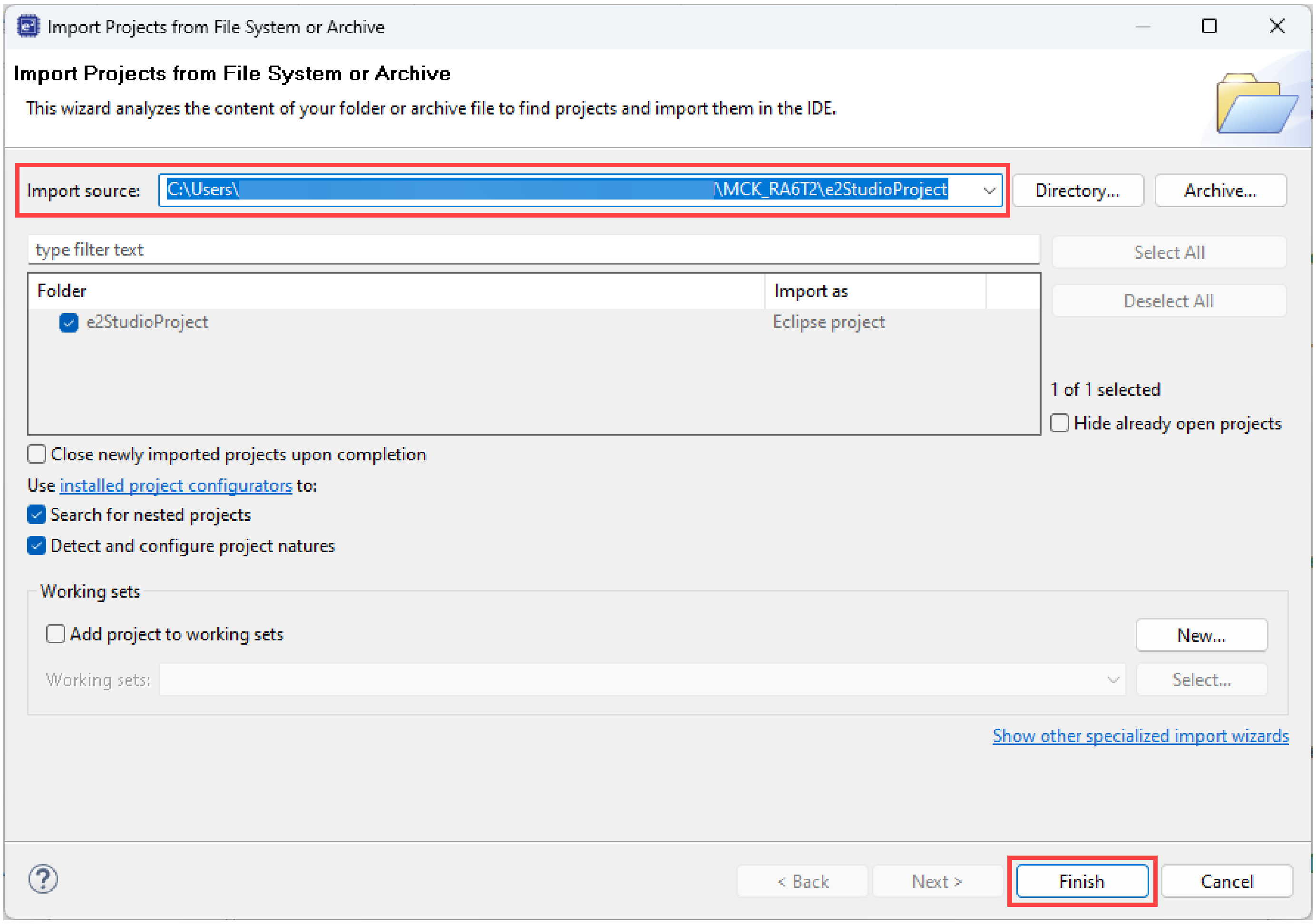
Deploy V/F Control Algorithm Code and Drivers to Hardware

Follow these steps to deploy the generated V/F control algorithm code along with hardware drivers to the MCK-RA6T2 motor control kit using the software e^2 studio:

1. Complete the hardware connections for the MCK-RA6T2 motor control kit (RTK0EMA270S00020BJ) and connect the kit to your computer. For details, see MCK-RA6T2 User's Manual.
2. Open the e^2 studio software.
3. Use the menu **File** > **Open Projects from File System** to open the **Import Projects from File System or Archive** window.

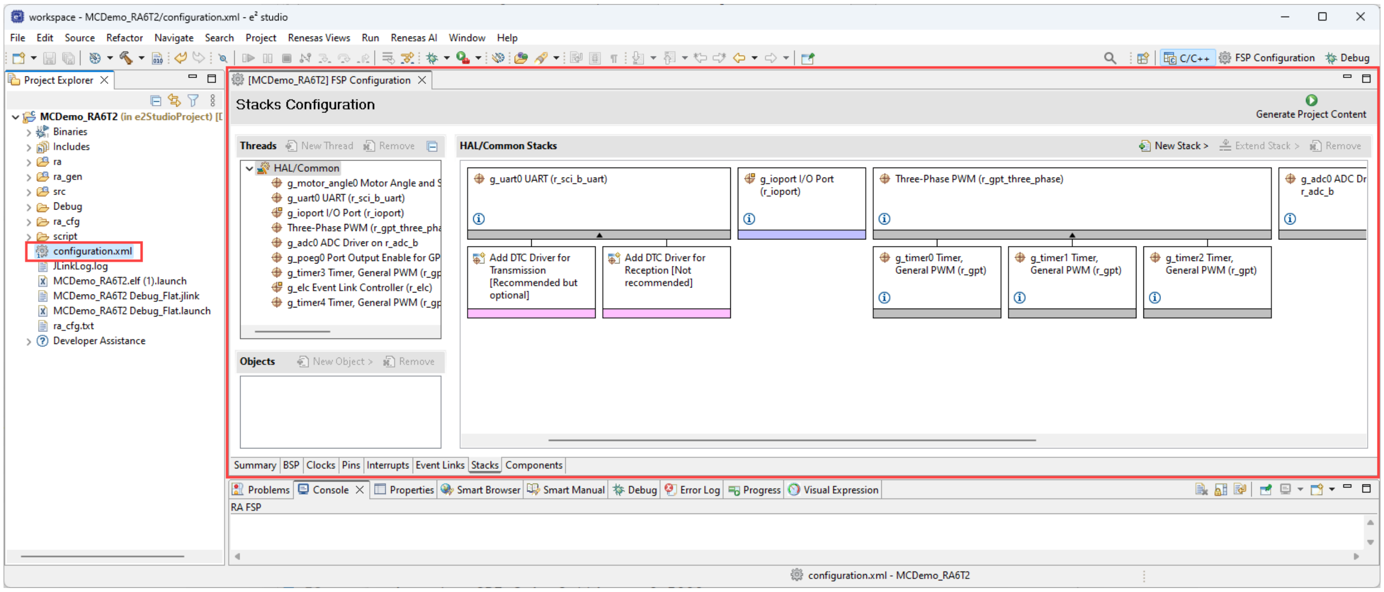


4. Use the **Import source** field to navigate to the `e2StudioProject` directory available in the unzipped folder `MCK_RA6T2` and then click **Finish**.

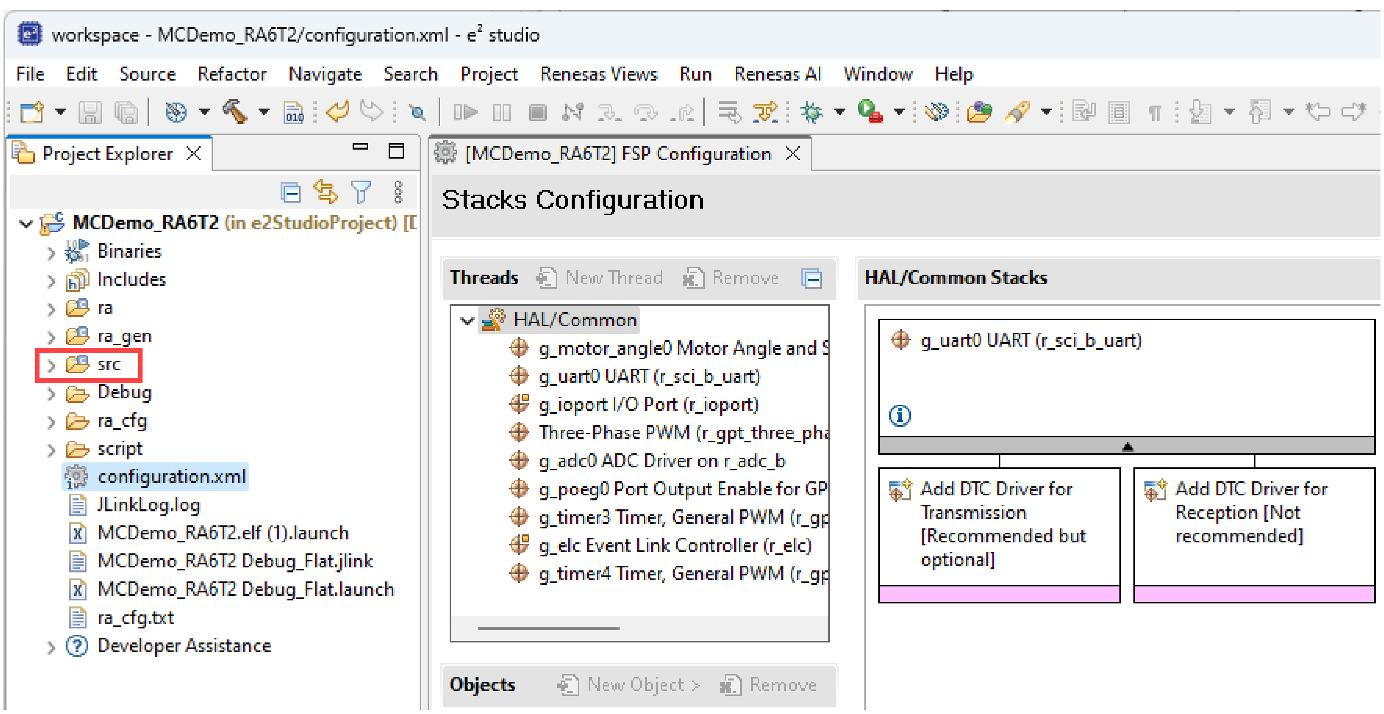


The software opens the Renesas hardware project for MCK-RA6T2 motor control kit in the **Project Explorer** panel.

The driver package for the MCK-RA6T2 motor control kit is configured in the file `configuration.xml` as shown in the following figure.

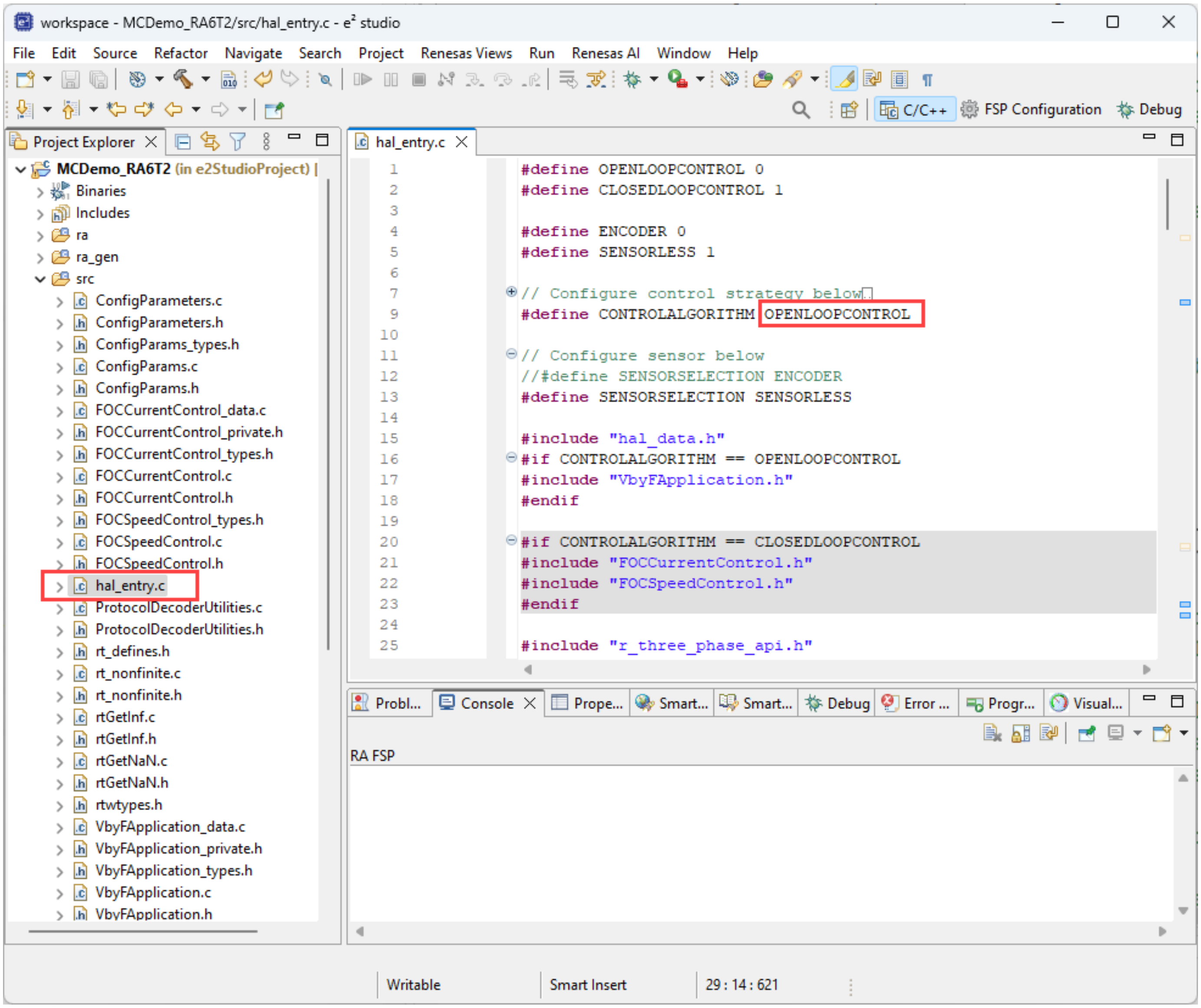


5 In the **Project Explorer** panel, expand the folder `src` to open its contents.



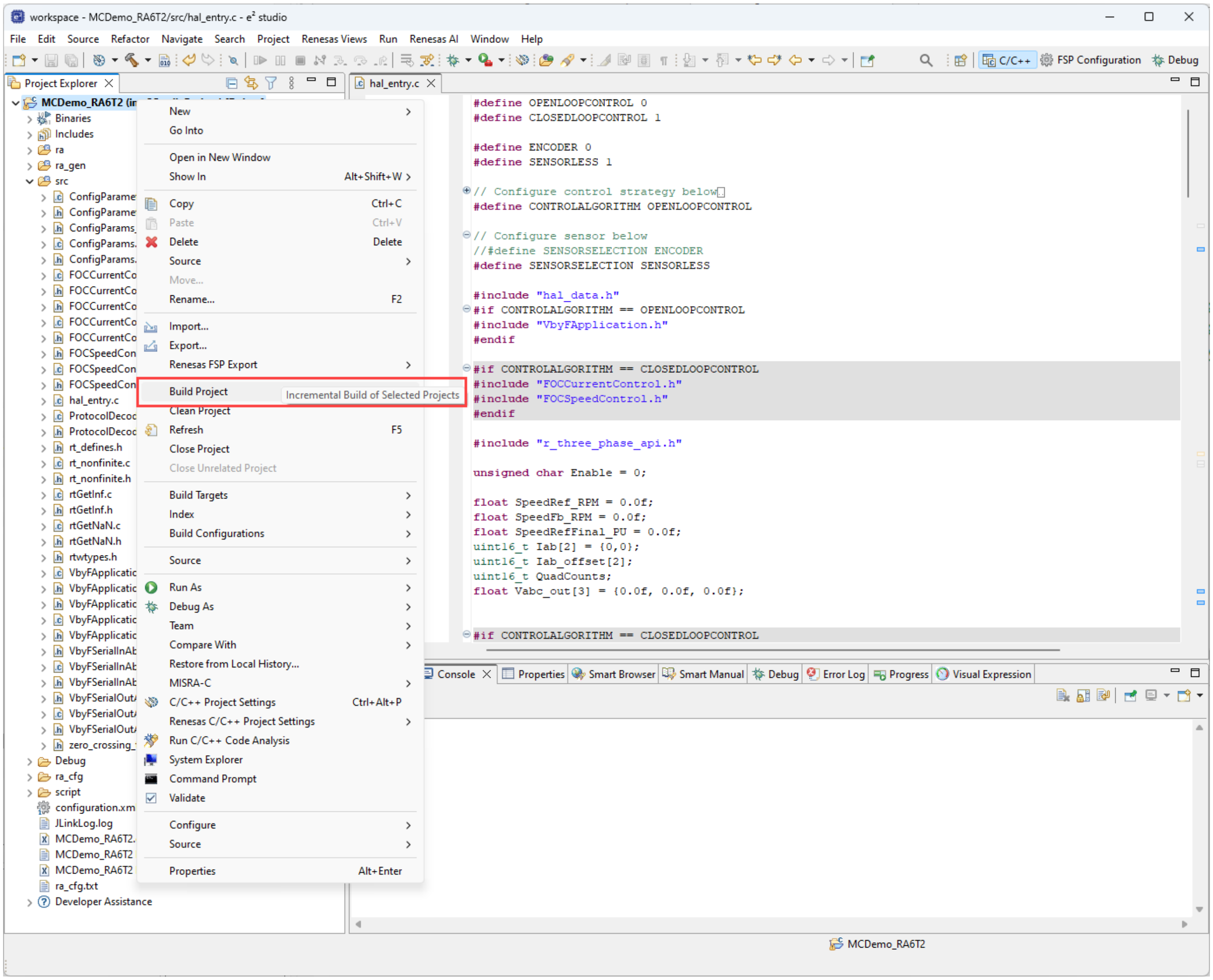
6 In the `src` folder, double-click the file `hal_entry.c` to open it in the editor.

7 Set the macro `CONTROLALGORITHM` to `OPENLOOPCONTROL` as shown in the following figure.

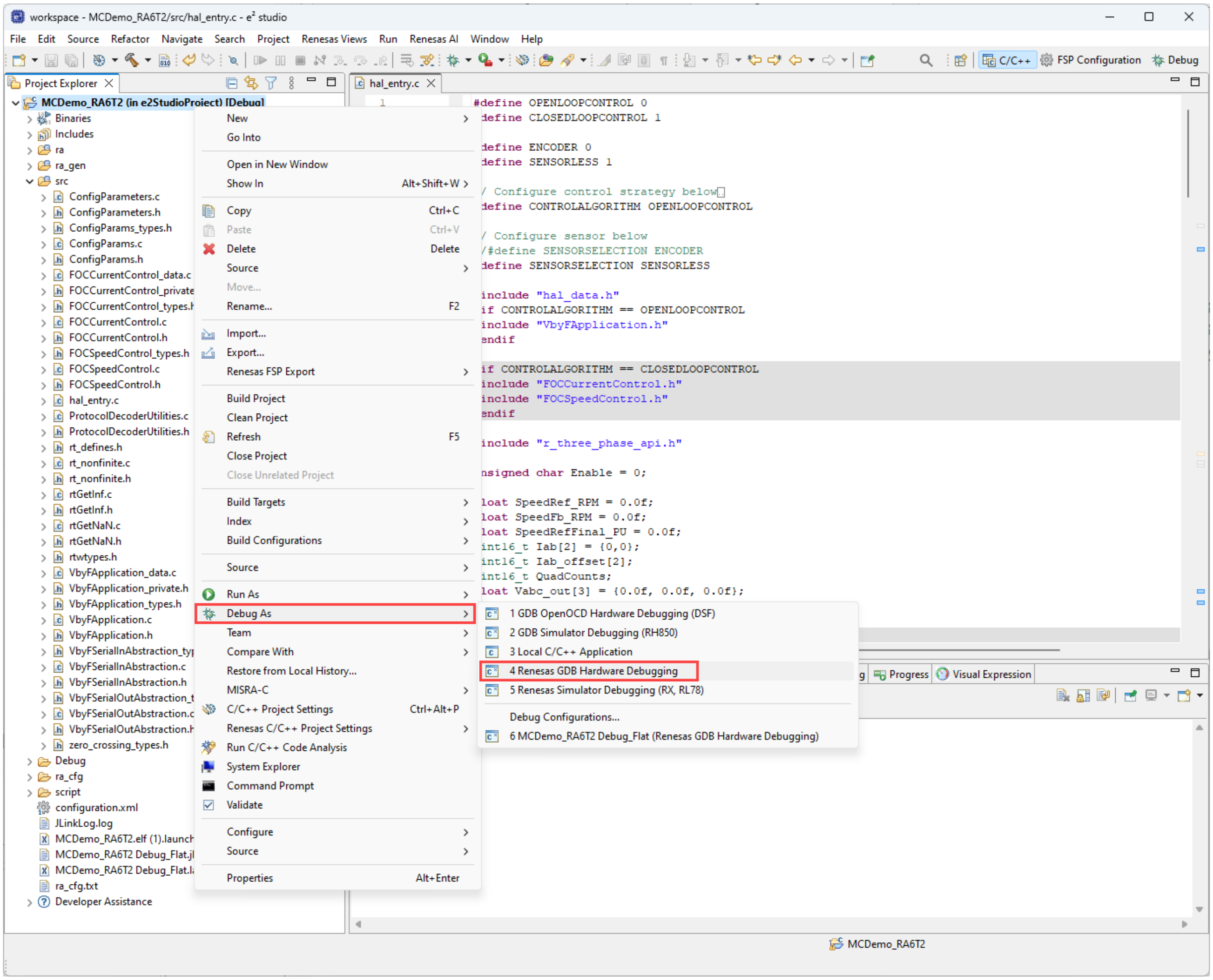


Save the file `hal_entry.c`.

8 In the **Project Explorer** panel, right-click the top-level folder `MCDemo_RA6T2` and select **Build Project** to build object code for the V/F control algorithm and hardware drivers.

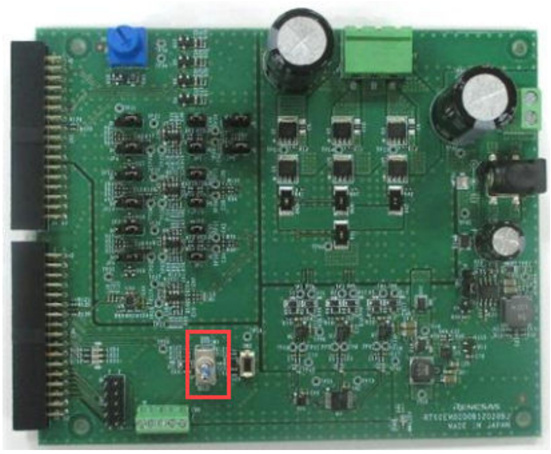


9 In the **Project Explorer** panel, right-click the top-level folder `MCDemo_RA6T2` and select **Debug As > 4 Renesas GDB Hardware Debugging** to deploy the object code for the V/F control algorithm and hardware drivers to the MCK-RA6T2 motor control kit.

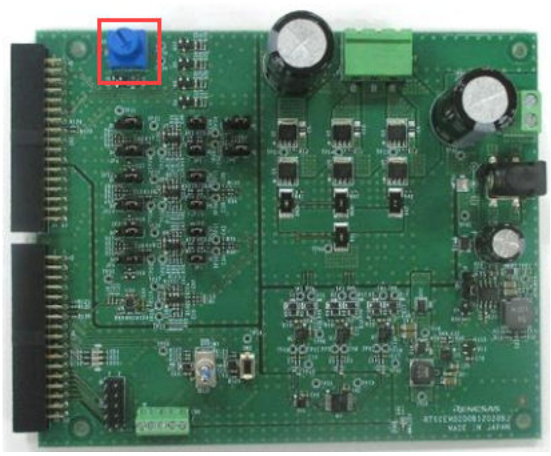


10 In the e2 studio software toolbar, click the **Run** button to start running the deployed code.

11 On the inverter board, turn the toggle switch to ON position to start running the motor.



12 During code execution on hardware, you can use the following pot (or dial) on the inverter board to change the reference speed for the V/F control algorithm.



13 To stop running the motor turn the toggle switch on the inverter board to OFF position.

Note

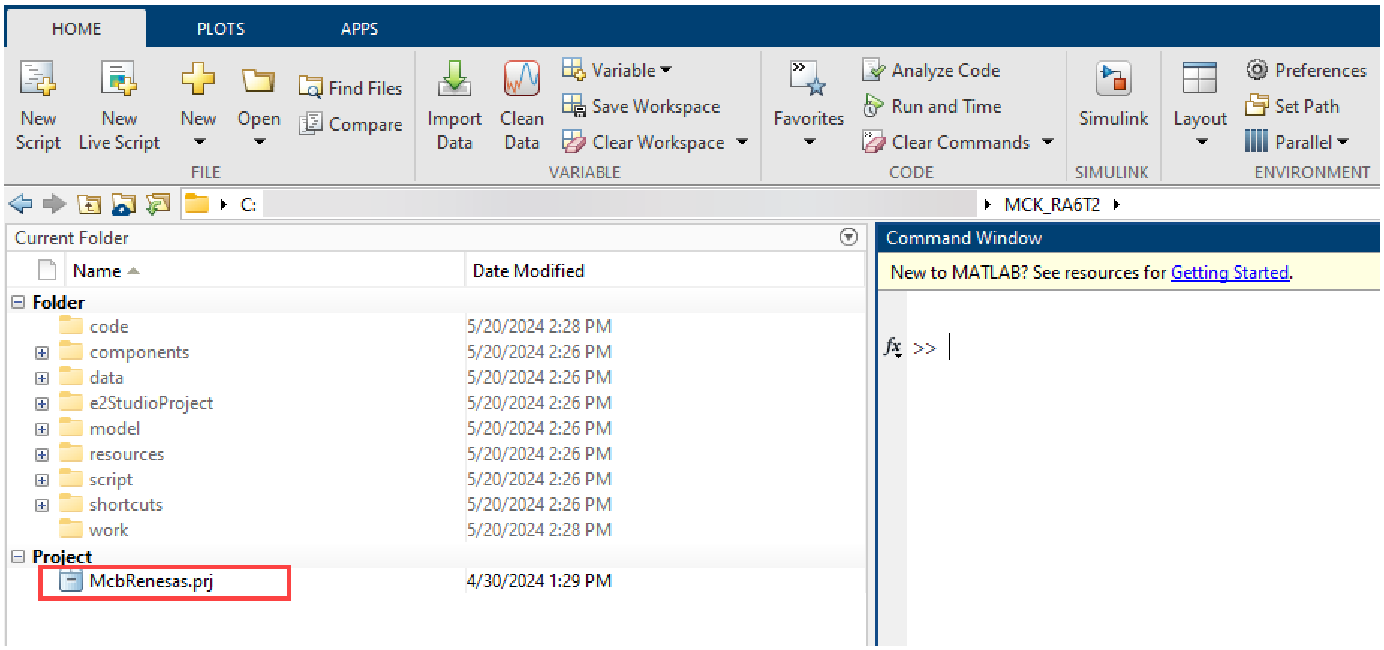
- You can safely disconnect the MCK-RA6T2 motor control kit from the computer after completing step 10.
- After you disconnect the kit from the computer, the deployed code stays on the kit and resumes execution even when you disconnect and reconnect the kit power supply.

Field-Oriented Control (FOC) of PMSM Using Renesas Hardware

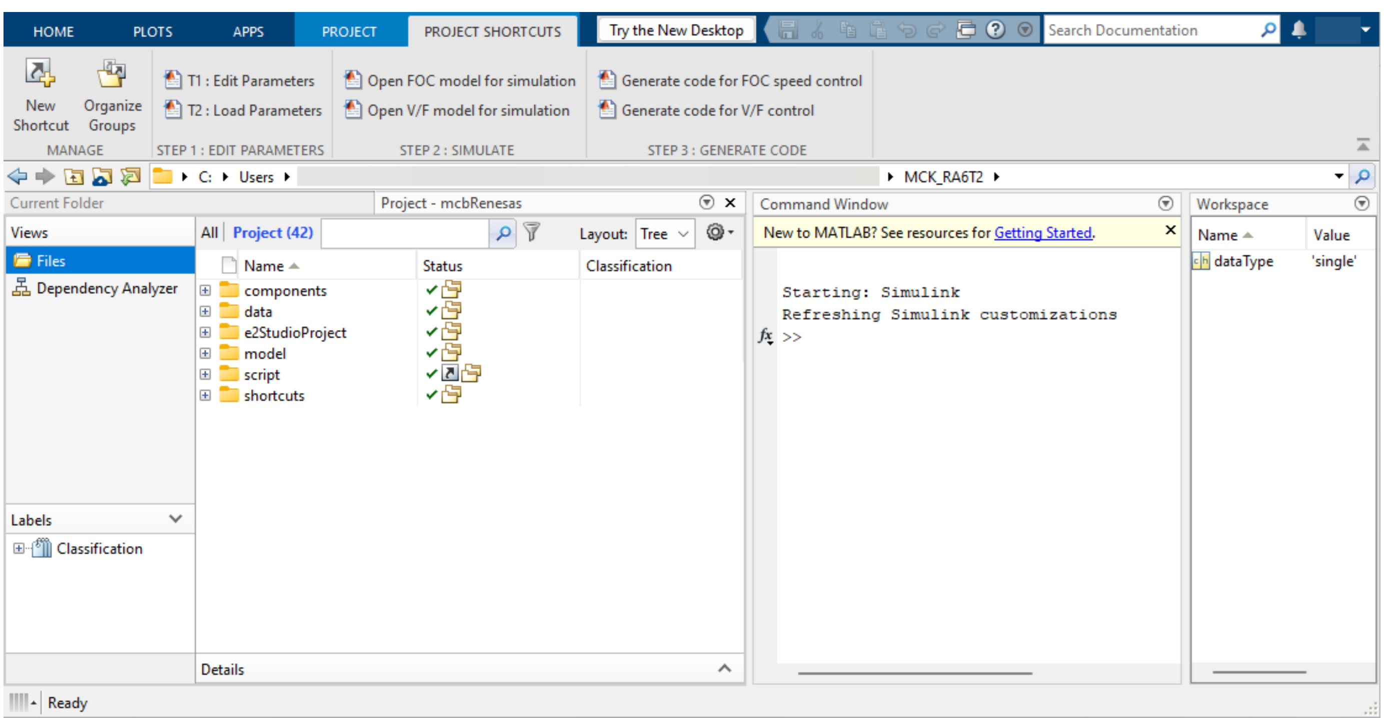
Open MATLAB Project

Follow these steps to open the MATLAB project associated with this example:

1. Unzip the `MCK_RA6T2.zip` folder that you downloaded from the GitHub repository.
2. Open MATLAB and navigate to the folder you unzipped in step 1.



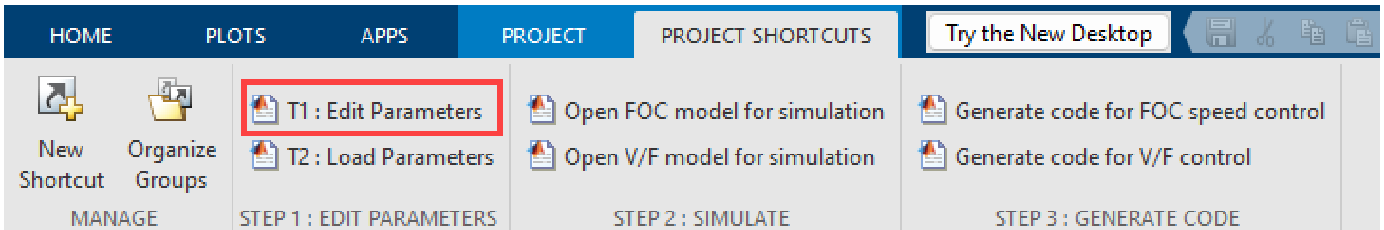
3. Double-click the file `McbRenesas.prj` to open the following MATLAB project window.



Update Motor and Inverter Parameters

The prerequisite to simulating or deploying the Simulink model containing FOC algorithm is to edit the motor and inverter parameters and load them to MATLAB base workspace. Follow these steps to update and load the parameters:

1. In the **Project Shortcuts** tab of the MATLAB project window, click the **T1: Edit Parameters** shortcut button to open the model initialization script `EnterParameters.m`.



   The Simulink model uses this script to obtain the motor, inverter, and other parameter values, which it uses to compute the derived parameters.

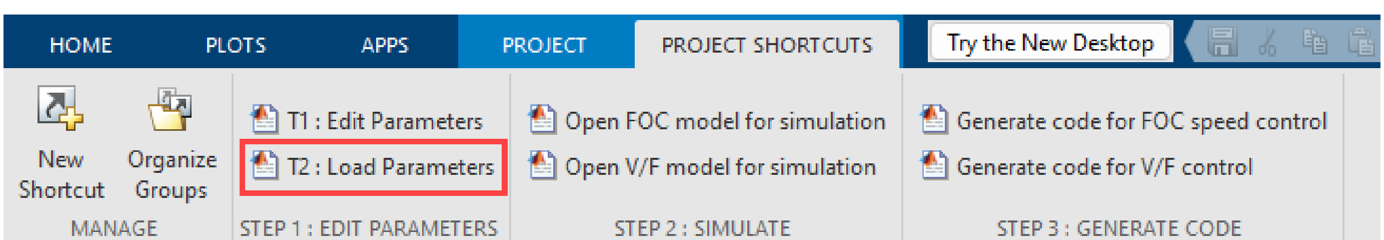
2. Update the following sections of the model initialization script `EnterParameters.m`:

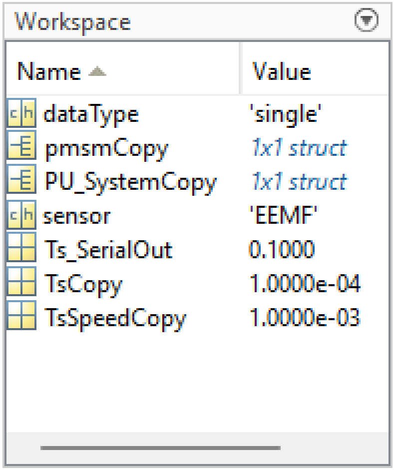
   - **Control Loop Frequencies**
   - **Inverter Parameters**
   - **Motor Parameters**

```
%% Control loop frequencies
fCurrentControl          = 10e3;     % Hz
fSpeedControl            = fCurrentControl/10;      % Hz

target.PWM_frequency     = 10e3;     % Hz           % The frequency configuration must be done in driver code too

%% Inverter Parameters
inverter.V_dc            = 24;       % V
inverter.ISenseMax       = 8.25;     % A            % Maximum current sensed by inverter
inverter.R_board         = 0.0033;   % ohm          % Rds_ON + Rshunt/3

%% Motor Parameters
pmsm.p                   = 4;        %              % Pole pairs
pmsm.I_rated             = 1.6;      % A
pmsm.N_rated             = 3500;     % RPM
pmsm.T_rated             = 0.1;      % Nm
pmsm.QEPSlits            = 1000;     %              % Quadrature encoder slits
pmsm.Rs                  = 0.75;     % ohm
pmsm.Ld                  = 0.5e-3;   % H
pmsm.Lq                  = 0.5e-3;   % H
pmsm.Ke                  = 7;        % Vpk_LL/kRPM
pmsm.J                   = 8e-6;     % Kg-m2
pmsm.B                   = 8e-5;     % Kg-m2/s
pmsm.V_boost             = 0.15;     % PU volt

%% Sensor selection
sensor = 'EEMF';                     % 'Encoder' , 'EEMF'

%% Derived Parameters
T_pwm = 1/target.PWM_frequency;
Ts = 1/fCurrentControl;
Ts_speed = 1/fSpeedControl;
Ts_motor = Ts/2;

Ts_SerialOut = 0.1;                                 % Required only for UART

pmsm.FluxPM    = (pmsm.Ke)/(sqrt(3)*2*pi*1000*pmsm.p/60);
pmsm.N_base    = mcb_getBaseSpeed(pmsm,inverter);
PU_System      = mcb_SetPUSystem(pmsm,inverter);
PI_params      = mcb_SetControllerParameters(pmsm,inverter,PU_System,T_pwm,Ts,Ts_speed);

% Create copies for EEMF block as the parameters are not tunable
```

3. Click **Save** to save the file `EnterParameters.m`.
4. Switch to the MATLAB project window and navigate to the **Project Shortcuts** tab.
5. Click the **T2: Load Parameters** shortcut button to:

   - Compute derived parameters using the data available in the model initialization script.
   - Update all parameters to the Simulink data dictionary `motorControlData.sldd`.
   - Load necessary parameters from Simulink data dictionary to the MATLAB base workspace.





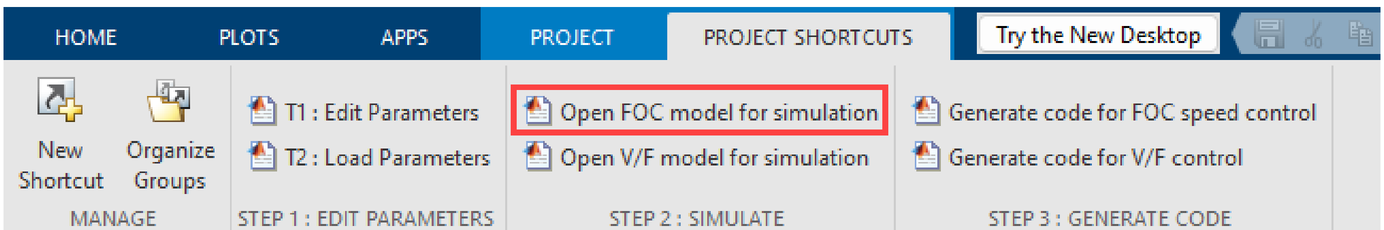
Note The shortcut **T2: Load Parameters** loads only the necessary parameters to the base workspace. The Simulink model reads the other parameters directly from the Simulink data dictionary.

Simulate Model Containing FOC Algorithm

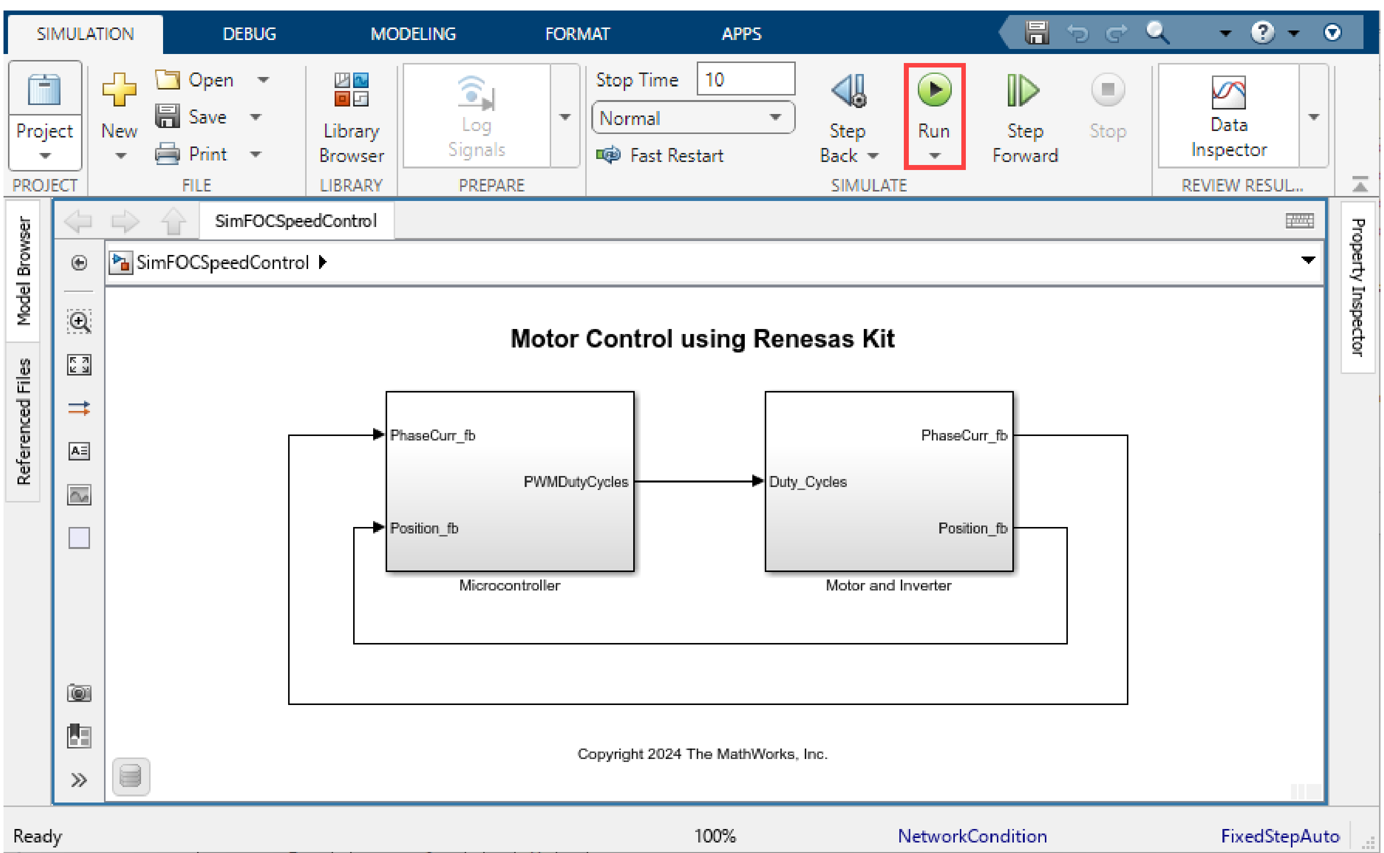
This example supports simulation. Follow these steps to simulate the Simulink model containing the FOC speed control algorithm.

1 Ensure that you edit the motor and inverter parameters and load them to the MATLAB base workspace. For details, see the section Update Motor and Inverter Parameters.

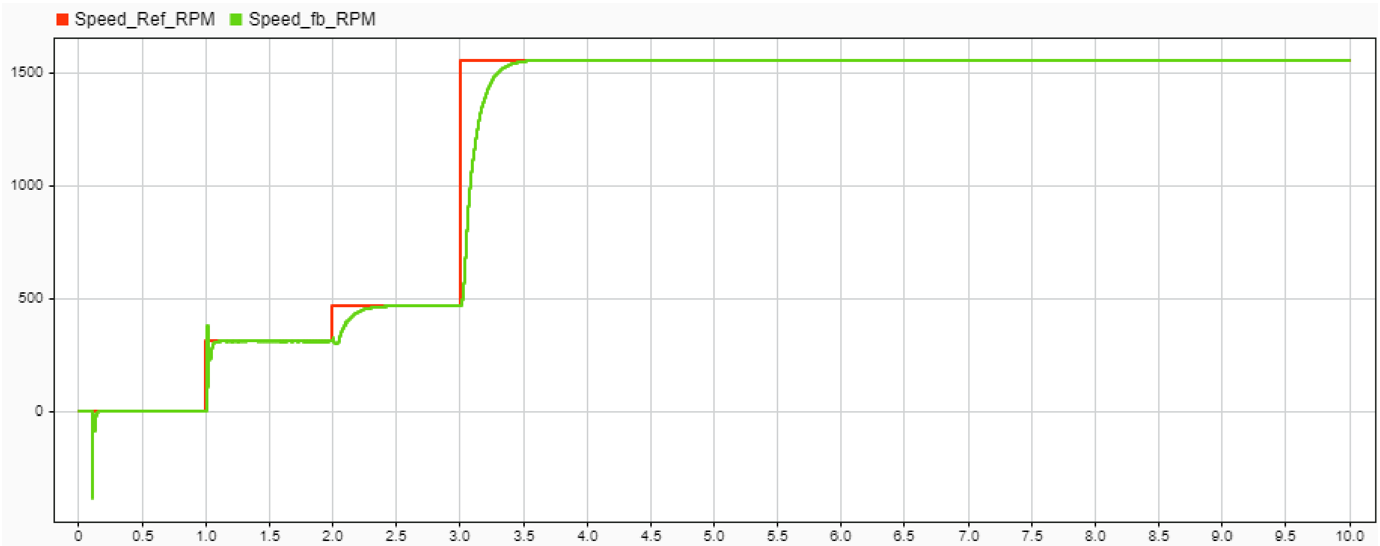
2 In the **Project Shortcuts** tab of the MATLAB project window, click the **Open FOC model for simulation** shortcut button to open the Simulink model `SimFOCSpeedControl.slx` containing the FOC speed control algorithm.



3 Click **Run** on the **Simulation** tab to simulate the model.



4 Click **Data Inspector** on the **Simulation** tab to view and analyze the simulation results.

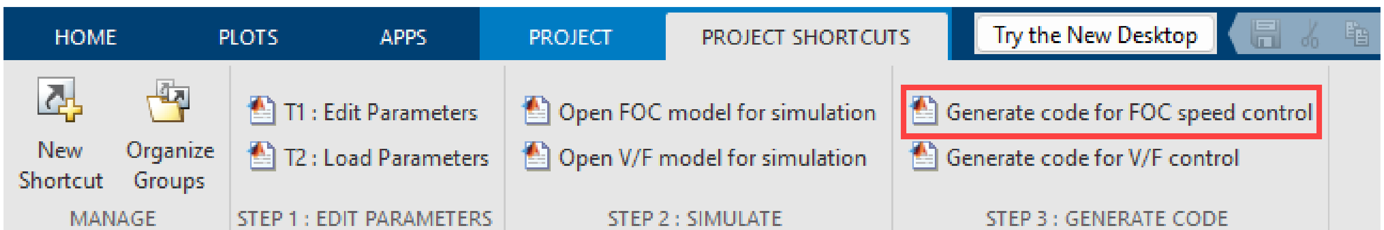

Speed_Ref_RPM
Speed_fb_RPM
1500
1000
500
0
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0

Generate Code for FOC Algorithm

This section explains how to use the Simulink model `SimFOCSpeedControl.slx` to generate code for the FOC speed control algorithm.

Follow these steps to generate the code:

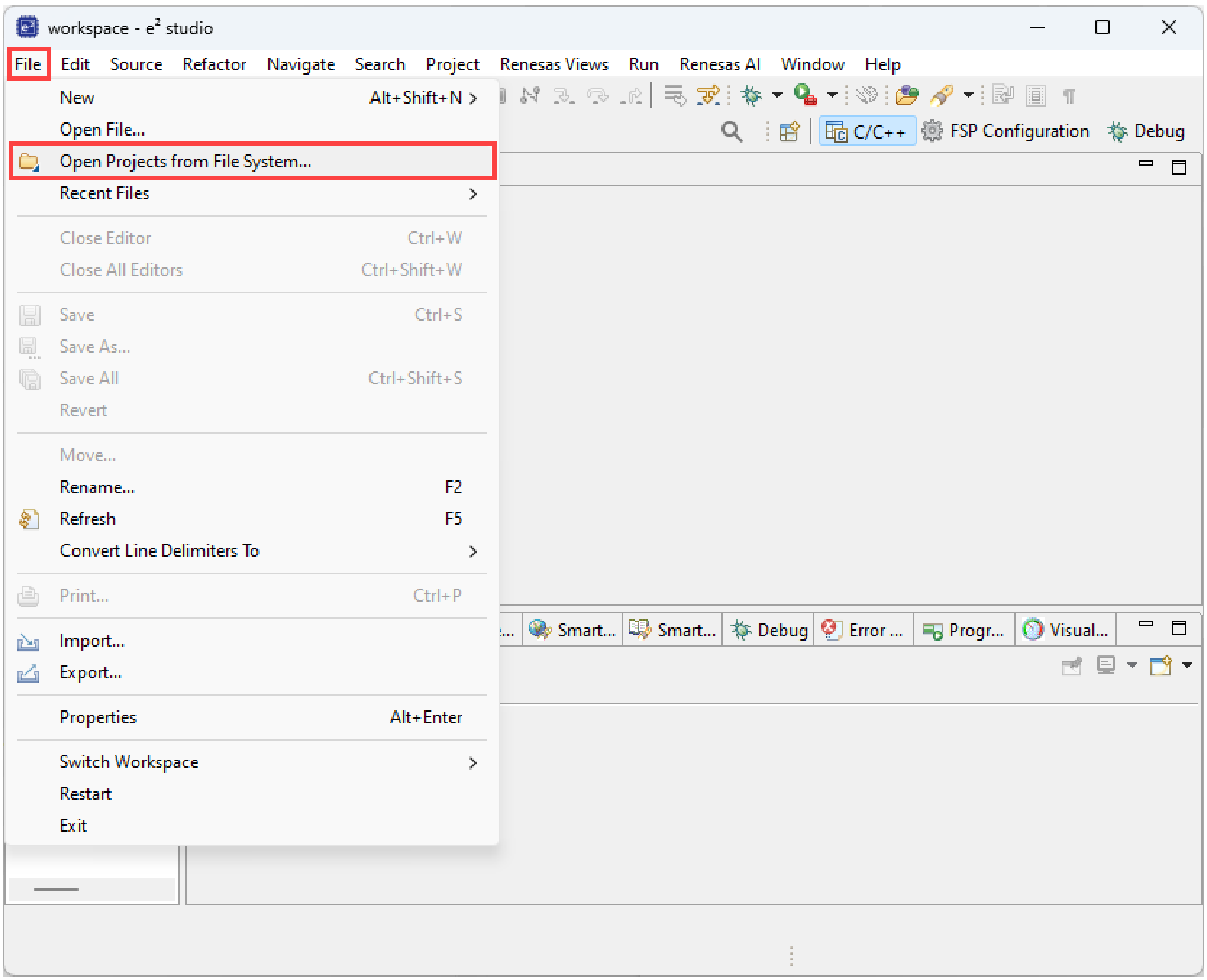
1. Simulate the model `SimFOCSpeedControl.slx` and observe the simulation results. For details, see the section Simulate Model Containing FOC Algorithm.
2. Click the **Generate code for FOC speed control** shortcut button to build the model `SimFOCSpeedControl.slx` as well as generate and save code inside `src` directory available in the `e2StudioProject` folder.



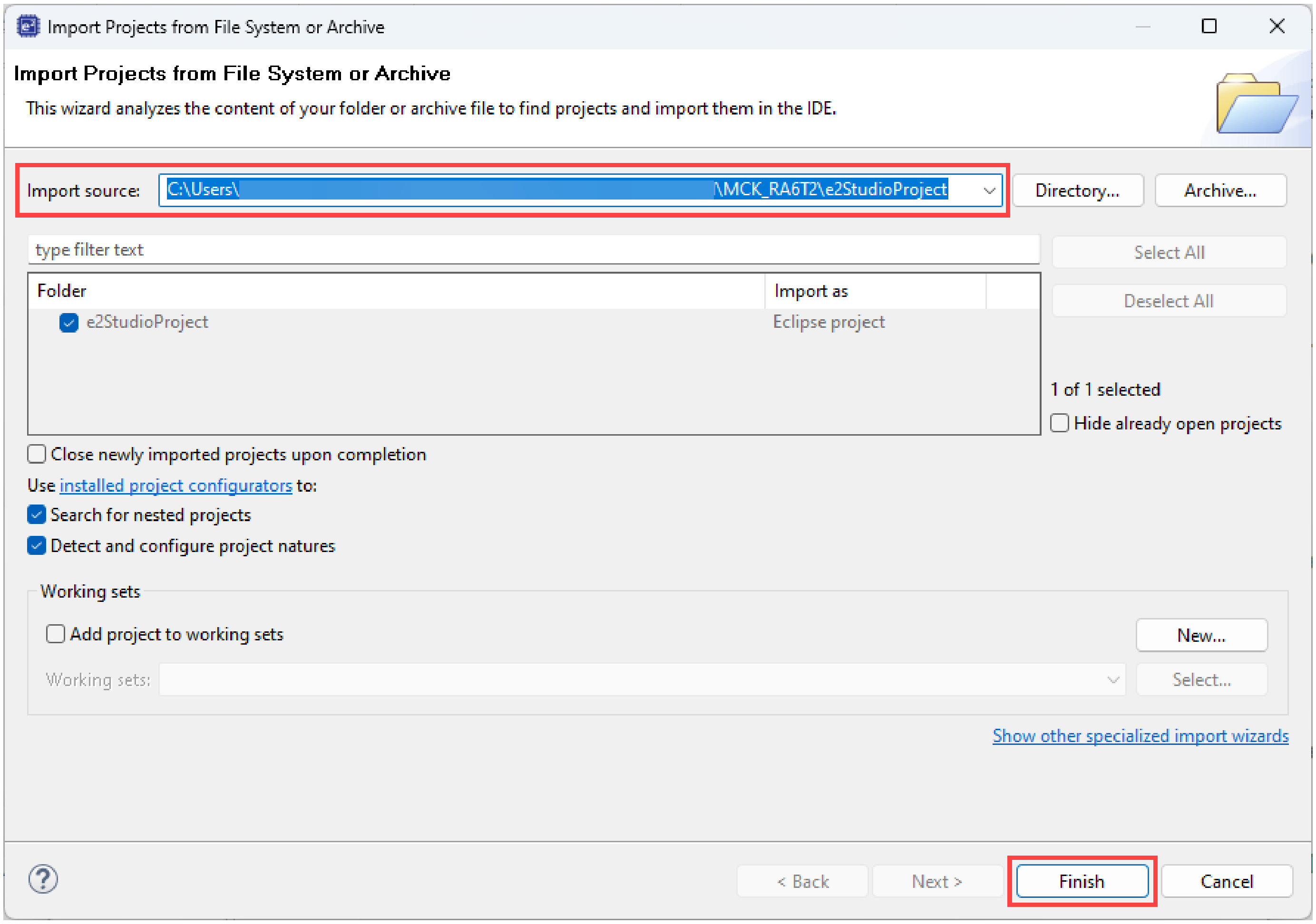
Deploy FOC Algorithm Code and Drivers to Hardware

Follow these steps to deploy the generated FOC speed control algorithm code along with hardware drivers to the MCK-RA6T2 motor control kit using the software e^2 studio:

1. Complete the hardware connections for the MCK-RA6T2 motor control kit (RTK0EMA270S00020BJ) and connect the kit to your computer. For details, see MCK-RA6T2 User's Manual.
2. Open the e^2 studio software.
3. Use the menu **File > Open Projects from File System** to open the **Import Projects from File System or Archive** window.

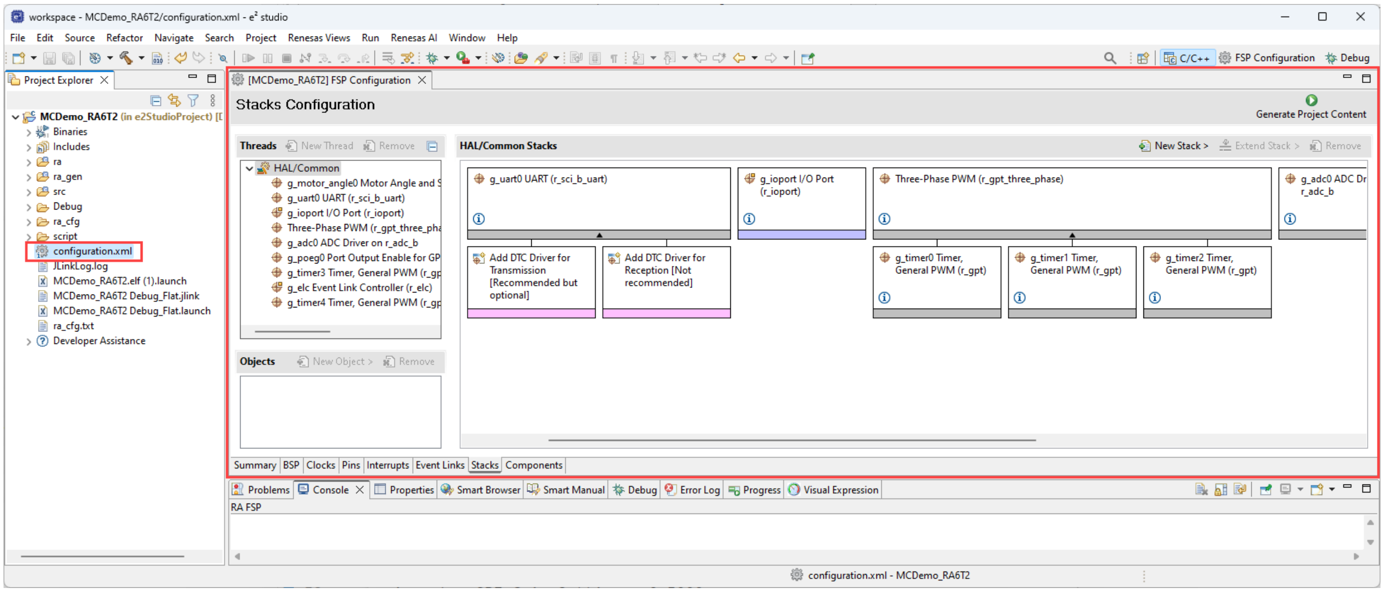


4. Use the **Import source** field to navigate to the `e2StudioProject` directory available in the unzipped folder `MCK_RA6T2` and then click **Finish**.

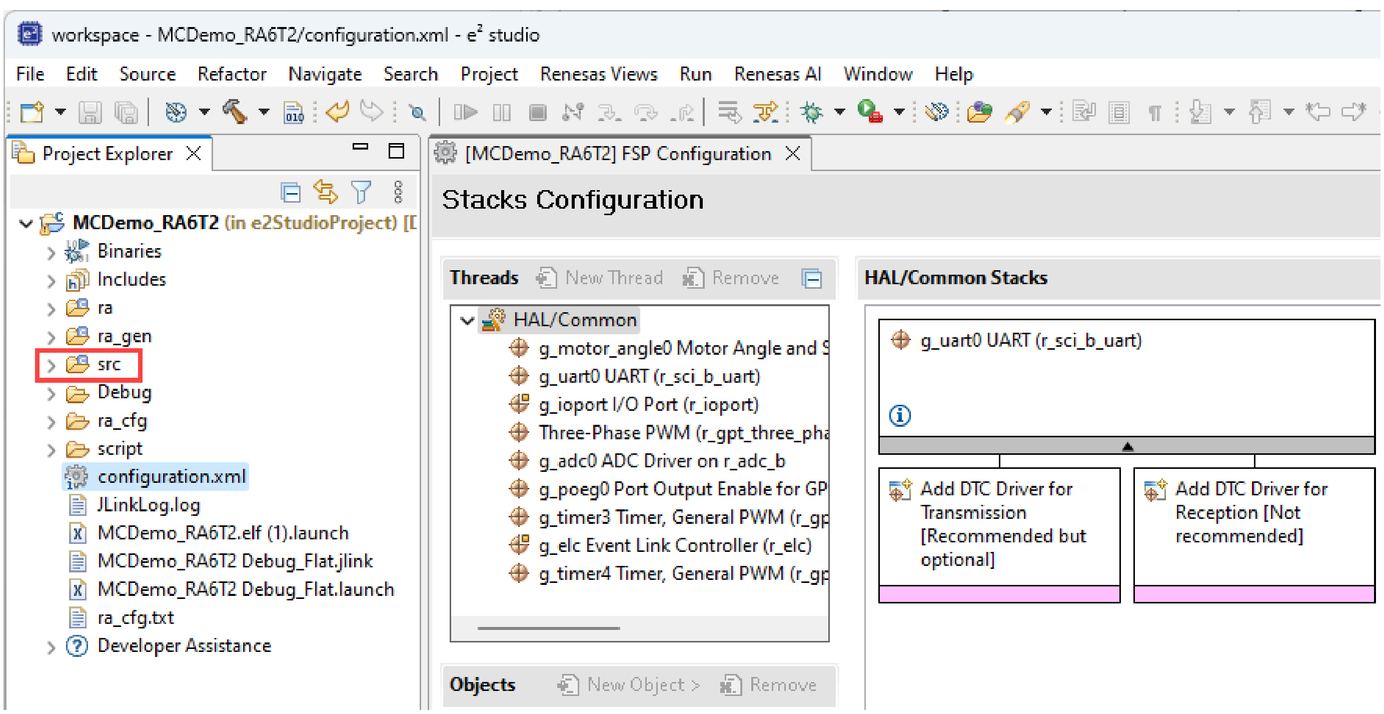


The software opens the Renesas hardware project for MCK-RA6T2 motor control kit in the **Project Explorer** panel.

The driver package for the MCK-RA6T2 motor control kit is configured in the file `configuration.xml` as shown in the following figure.

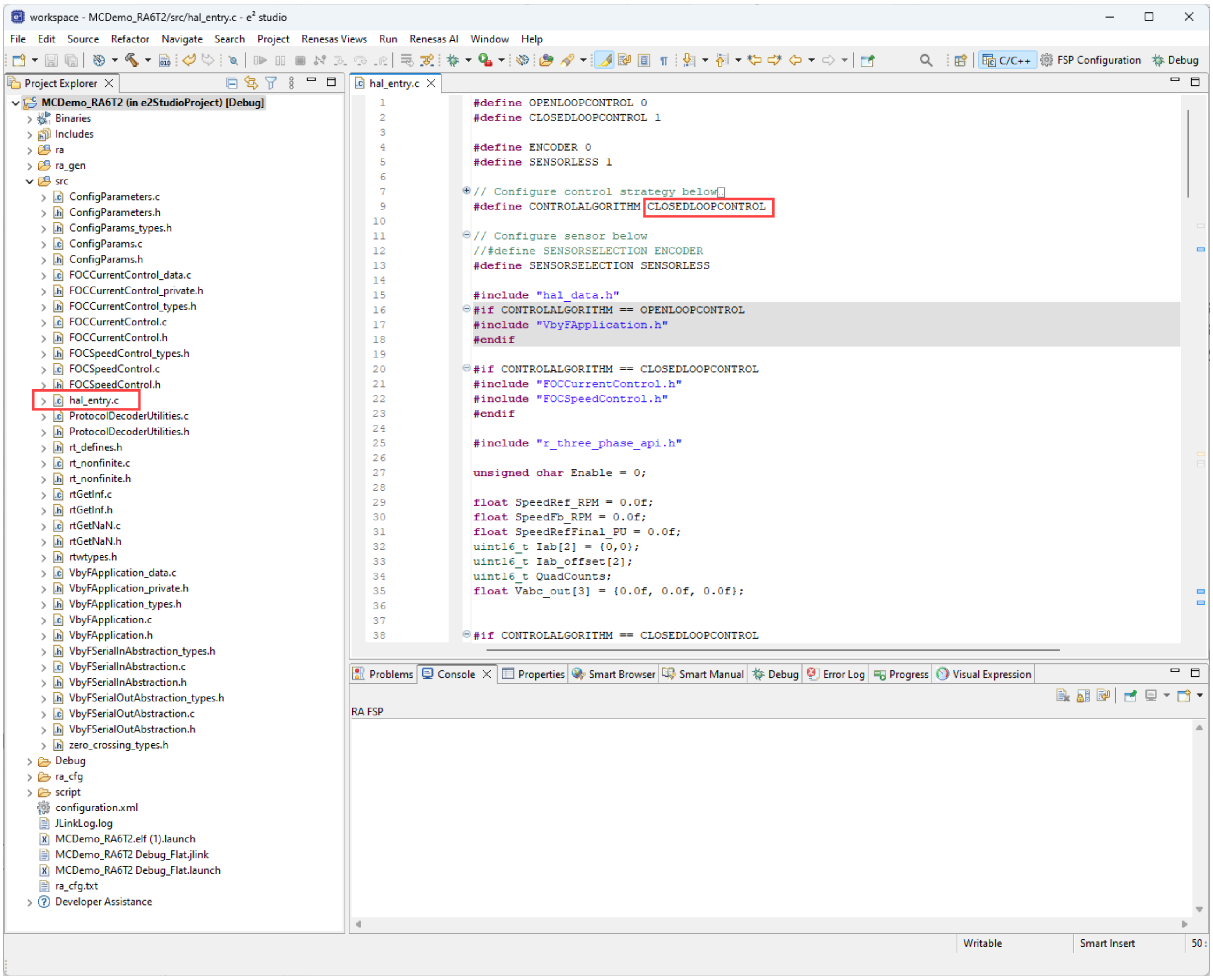


5 In the **Project Explorer** panel, expand the folder `src` to open its contents.



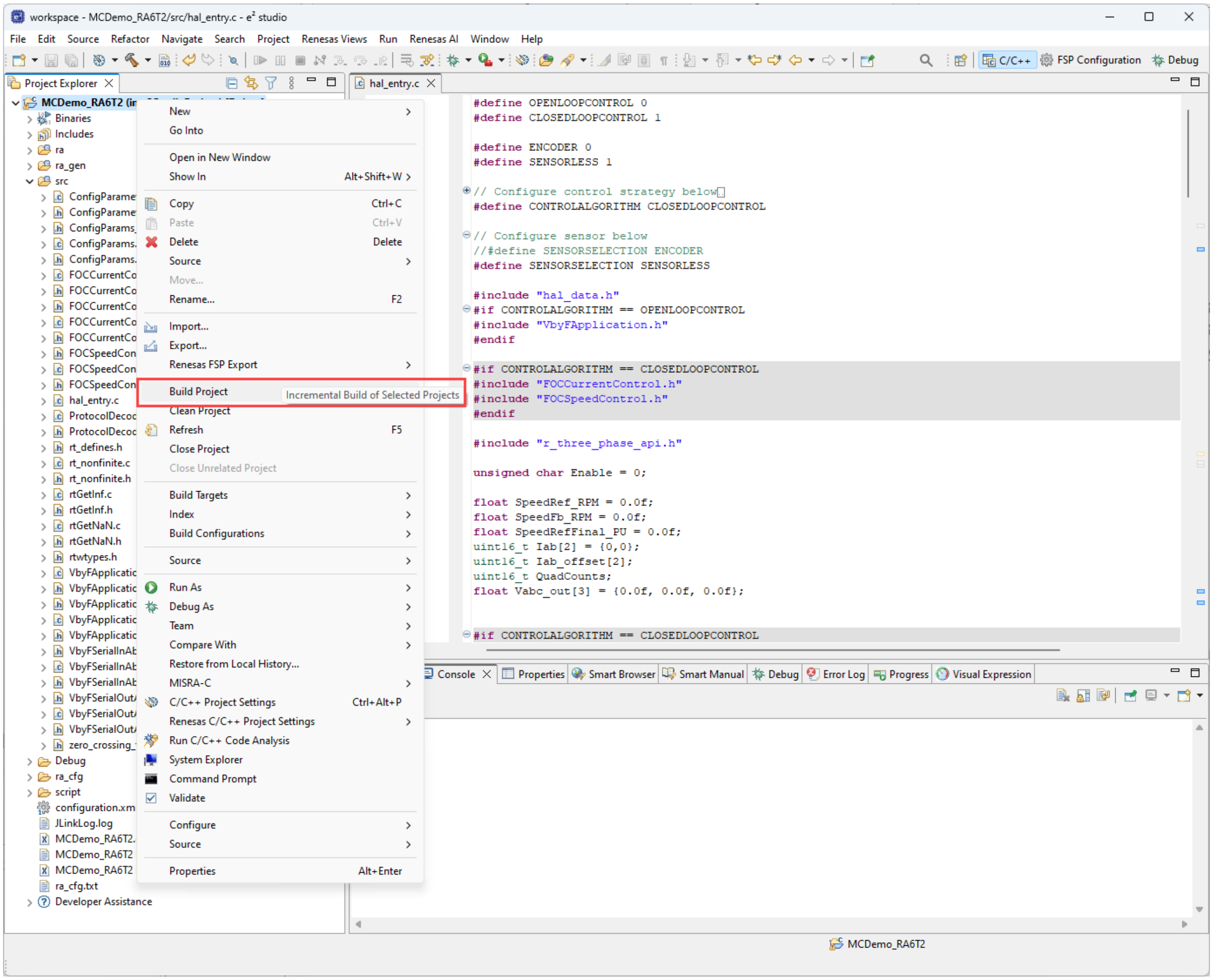
6 In the `src` folder, double-click the file `hal_entry.c` to open it in the editor.

7 Set the macro `CONTROLALGORITHM` to `CLOSEDLOOPCONTROL` as shown in the following figure.

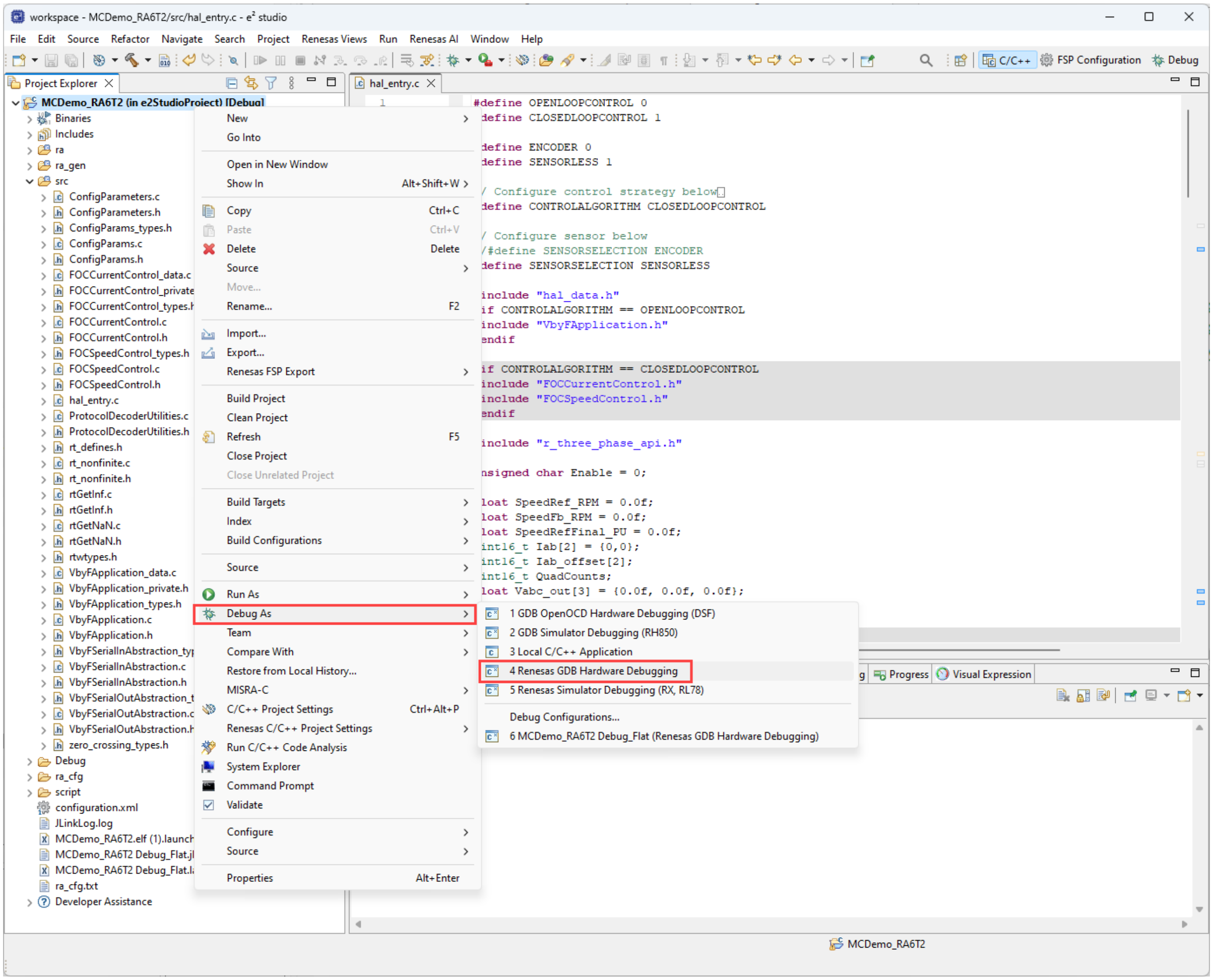


Save the file `hal_entry.c`.

8 In the **Project Explorer** panel, right-click the top-level folder `MCDemo_RA6T2` and select **Build Project** to build object code for the FOC algorithm and hardware drivers.

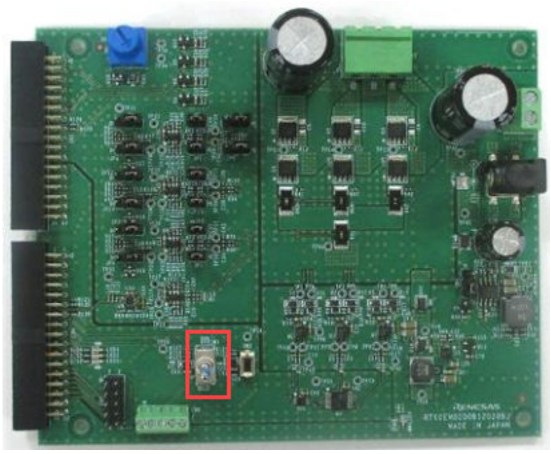


9 In the **Project Explorer** panel, right-click the top-level folder `MCDemo_RA6T2` and select **Debug As** > **4 Renesas GDB Hardware Debugging** to deploy the object code for the FOC algorithm and hardware drivers to the MCK-RA6T2 motor control kit.

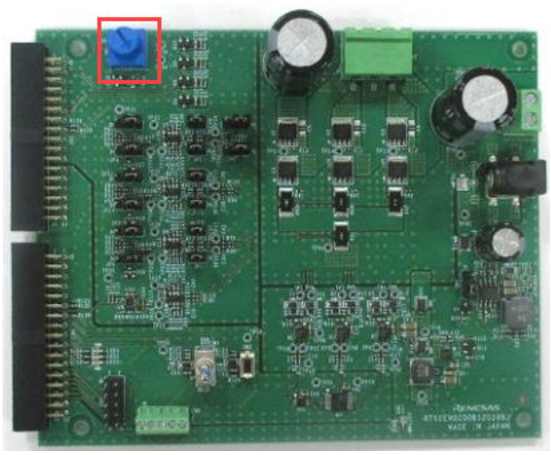


10 In the e2 studio software toolbar, click the **Run** button to start running the deployed code.

11 On the inverter board, turn the toggle switch to ON position to start running the motor.



12 During code execution on hardware, you can use the following pot (or dial) on the inverter board to change the reference speed for the FOC algorithm.



13 To stop running the motor turn the toggle switch on the inverter board to OFF position.

Note

- You can safely disconnect the MCK-RA6T2 motor control kit from the computer after completing step 10.
- After you disconnect the kit from the computer, the deployed code stays on the kit and resumes execution even when you disconnect and reconnect the kit power supply.
